EXHIBIT 9

BladeCenter chassis management

T. Brey
B. E. Bigelow
J. E. Bolan
H. Cheselka
Z. Dayar
J. M. Franke
D. E. Johnson
R. N. Kantesaria
E. J. Klodnicki
S. Kochar
S. M. Lardinois
C. A. Morrell
M. S. Rollins
R. R. Wolford
D. R. Woodham

The IBM eServer™ BladeCenter® system allows compute, network, and storage components to operate under a common chassis management scheme. It offers a new approach to solving many systems management issues surrounding the integration of power, packaging, cooling, media peripherals, and cabling of multiple compute servers, network switches, and storage into one highly redundant package. In the BladeCenter design, these functions are integrated into the chassis, allowing the costs of each shared component to be amortized across the entire chassis. At the heart of the system is the management module hardware and firmware that provides chassis management for all components, thereby removing the cost and complexity of having to manage each component independently.

Introduction

This paper describes how chassis management is a multi-tiered management concept centered around a management module that provides control over components such as blades, input/output (I/O) switch modules, and blowers within the IBM eServer* BladeCenter* system. To understand the rationale for the functions provided by the management module and its associated hardware, one must have a good understanding of the components being managed. Details on each component can be found in [1].

Before the advent of the BladeCenter system, IBM enterprise management software products such as Director or Tivoli* were attached to servers via low-cost, low-functioning interfaces such as Alert Standard Format [2]. In addition, server vendors offered optional higher-cost, higher-functioning management controllers with network interfaces, such as the IBM Remote Supervisor Adapter [3, 4]. Although adding full-featured network manageability came at a higher price, more and more customers found themselves placing servers in remote locations where there were few if any people with the IT skills required to keep servers functioning for long periods of time. Many found that the cost of remote management was much less expensive than making a single trip to the remote location.

In the BladeCenter design, each chassis contains a very robust and network-aware management module that provides a consistent administration interface for all components in the chassis: 14 blades, four I/O switch modules, and common chassis components such as a keyboard, video, mouse (KVM), media tray, and the management module itself. The cost of the management module hardware is amortized over all BladeCenter components, which reduces the overall cost of systems management in the chassis. The management module exposes those functions through a number of external interfaces, discussed below, and acts as an aggregation point for administration of all components within the chassis. As new components are developed, standard functions such as power controls, environmental alerting, KVM controls, and so on remain consistent, avoiding changes to enterprise management software due to new components in the chassis.

The management module (MM) provides for remote management of the chassis by enterprise managers such as IBM Director [5] or Tivoli, or by other enterprise management software including point-to-point access via a Web browser, Telnet or Secure Shell (SSH) client, or Simple Network Management Protocol (SNMP) manager [6]. Service processors—i.e., the processor blade baseboard management controller (BMC) or the I/O switch module control point—within each component supply local management on the components in which they reside. The MM is the intelligent control point for shared chassis resources such as power modules [7],

0018-8646/05/$5.00 © 2005 IBM

blowers, compact disk read-only memory (CD-ROM), floppy disk, Universal Serial Bus (USB) port, or KVM. Service processors such as the MM have been designed into IBM xSeries* and eSeries servers for many generations. Many of the features and functions of the MM are built on top of functions that already exist and are provided by the IBM Remote Supervisor Adapter (RSA) [3, 4], an optional component for many xSeries servers. When customers needed full-function remote management, they used the RSA. At the time the BladeCenter project was conceived, it was clear that most customers would prefer remote management of their servers if not for the additional cost of the RSA function. Because of the unique high-density BladeCenter packaging, a single management controller (i.e., the MM) could manage many servers, effectively reducing the cost of remote management on a per-server basis by 14 (the number of blades in a chassis). This made remote full-function systems management very affordable. The RSA remote management function was used as a reference design, enabling IBM to move quickly in developing the hardware and software needed to provide a rich set of system management functions on a flexible platform while meeting stringent time-to-market objectives.

This paper is organized to provide a high-level view of the hardware and firmware aspects of chassis management including the MM execution environment, BladeCenter redundancy, and the external and remote systems management interfaces. This is followed by a lower-level discussion of such chassis management functions as monitoring and error reporting, inventory management, configuration validation, power and thermal management, MM discovery, KVM controls, remote disk, serial over LAN (SOL), video, I/O module management, firmware updates, and chassis diagnostics.

Chassis management

BladeCenter chassis management defines an interface architecture that enables all components to communicate certain information to the MM and for the MM to control each component in the chassis. This communication begins when the module is plugged into the chassis or when ac power is applied to the chassis. There are two power domains in each component: power domain 1 (circuitry powered by the auxiliary or standby domain) and power domain 2 (full power to the component). When plugged into the chassis (or when ac power is applied), hot-swappable components signal the MM to indicate that they are present. At that time, the component powers all of the logic required to communicate with the MM from power domain 1. The MM interacts with all BladeCenter components using either the RS-485 bus or the Inter-Integrated Circuit (I^2C) Serial Bus Interface [8]. Each component identifies itself by responding to messages over the RS-485 interface or by I^2C bus access to nonvolatile storage that contains vital product data (VPD) fields. The MM checks the various components within the chassis to ensure that the requirements of each installed component will allow it to work properly with all other chassis components (e.g., power consumption, network switch port to processor blade port protocol mismatch). If these checks pass, power domain 2 of the component is allowed to be turned on. After the component runs diagnostic and power-on self-tests, the MM monitors the component for informational, warning, or failure events. These events are logged into the MM event log for later retrieval and can also be sent directly to a remote systems management application to inform the system administrator.

BladeCenter systems management also provides the ability to remotely administer a chassis. The external MM interfaces provide an administrator with various tools, policies, and procedures to manage all components. This paper describes below how the MM manages components within the chassis and how shared resources are allocated to each component.

Remote management of the entire chassis is implemented using both industry-standard and proprietary protocols. The MM enables management entities such as IBM Director to automatically discover each BladeCenter server in a network and expose out-of-band control over all components in a chassis; it allows systems management applications to gather information about the current state of each component. The MM provides services such as the ability to update firmware on various components in the system, remotely control each processor blade, and remotely mount drives for access by processor blades over an internal USB port.

Chassis hardware is fully redundant and comprises fully redundant MMs, redundant buses to each blade [9], I/O switch module [10], redundant buses on the midplane [11], and power module, tachometer control to each blower, and redundant Ethernet connections to all I/O switch modules. This hardware architecture enables duplicate MMs to provide highly available chassis management services.

Management module hardware works in conjunction with hardware on the midplane [11], processor blade, I/O switch module, media tray (i.e., CD-ROM, floppy, USB port), power module, and chassis blowers. The MM is the central control point providing all of the controls necessary to share the components mentioned above in a coordinated fashion. It does this using redundant RS-485, USB, Ethernet, and I^2C buses. The RS-485 bus allows communication between the MM and the local BMC on the processor blade. USB buses are used for the keyboard, mouse, media tray (MT), and remote disk functions. Multiple I^2C buses are used to communicate with all other I^2C bus devices in the chassis, including

WSOU-ARISTA00001297

devices on the midplane, the media tray, blowers, power modules, and I/O switch modules. In addition to the RS-485 and I^2C bus interfaces, internal MM-to-I/O-switch-module Ethernet ports are provided for external management of the I/O switch module and are used internally for blade SOL communications. Video from each blade can be displayed locally at the console attached to the MM. Video can also be displayed remotely over the network using hardware in the MM that packetizes the video stream and sends it to a remote user over an Ethernet network.

As shown in **Figure 1**, the MM is an embedded system that contains a 200-MHz PowerPC* (PPC) 405GP, 32 MB of error-correcting code (ECC) memory, 4 MB of flash containing PPC boot and application code, and 256 KB of nonvolatile random access memory (NVRAM) containing MM configuration and chassis state information.

Using the external Ethernet link, administrators anywhere on the private management network can remotely manage the BladeCenter chassis and all of its components. The MM contains two Ethernet network interface cards (NICs). One NIC is connected to the private management network through an RJ45 jack on the rear of the MM. The second NIC is connected to an internal five-port Ethernet switch located in the MM. Each of the four remaining ports is connected to each possible I/O switch module in the chassis. The Ethernet connection to each I/O switch module is used to provide the switch management applications running on the private management network with the ability to route across the MM to the I/O switch module, usually to monitor and configure the I/O switch module.

The MM contains two sets of RS-485 transceivers wired to the two RS-485 midplane buses. Each processor blade contains two RS-485 buses. The MM makes use of USB buses from each blade to expose a keyboard and mouse port from each processor blade. The media tray also makes use of USB ports to provide local storage to each processor blade. Each MM contains a connector to which a keyboard and mouse are attached.

The I^2C bus is the *de facto* standard bus used in embedded designs. It offers low cost and the ability to interconnect a wide assortment of devices. Redundant I^2C buses are used throughout (Figure 1).

Each MM provides a separate Ethernet port to each I/O switch module. Therefore, a management module failover would cause traffic to flow on a completely independent path.

The MM contains a front panel that contains the Ethernet port on the private management local area network (LAN), along with the KVM connections from which any one of the 14 processor blades can be controlled. The front panel also contains light-emitting diodes (LEDs) for Ethernet activity, MM power status, MM active, and MM fault indicators.

By design, the MM supports a fully redundant configuration. Hot-plugging of the MM can occur at any time without detriment to the remaining MM or the current state of the chassis.

The following sections describe the major functions of a BladeCenter chassis, beginning with a view into the MM execution environment, redundancy aspects, external network interfaces, monitoring, alerting, and other management functions. These functions present a rich systems management interface on behalf of all components in the BladeCenter chassis.

MM execution environment

At power on, the MM PowerPC processor executes a built-in self test (BIST) and then executes the basic I/O system (BIOS) to initialize MM hardware and verify that the MM components are functioning properly. BIOS checks for proper BIST results, selects one of two possible flash images from which to boot, and then loads the embedded operating system (OS). This OS is fully preemptive and task-based, and has a small (50 KB) footprint. The MM firmware and MM application firmware (written in C language and assembler) is made up of tasks that provide the autonomous services collectively known as *chassis management*.

Figure 2 is a high-level representation of the MM firmware. The OS kernel represents the embedded OS that is implemented on top of the MM hardware. The MM application firmware comprises the various tasks and services that are shown and are grouped into interface tasks, functional managers, chassis management, a protocol task layer, a physical interface driver layer, and hardware controllers.

The interface tasks support external connections into the MM from the private network. The functional managers describe various tasks that represent the behavior of the MM as viewed by a user. Chassis management represents behavior provided by the MM to support shared chassis resources such as power modules, blowers, I/O switch modules, and blades. The physical interface and protocol task layers represent chassis hardware interfaces that allow the MM to control all of the underlying components of the chassis. Finally, the MM hardware is made up of hardware controllers that assist in offloading specific processing from the PowerPC processor. These are described in more detail in subsequent sections of this paper.

BladeCenter redundancy

A BladeCenter system offers redundancy by duplicating pluggable modules or buses on the midplane. When multiple components exist in the chassis (processor blades, I/O switch modules, power modules, blowers, and

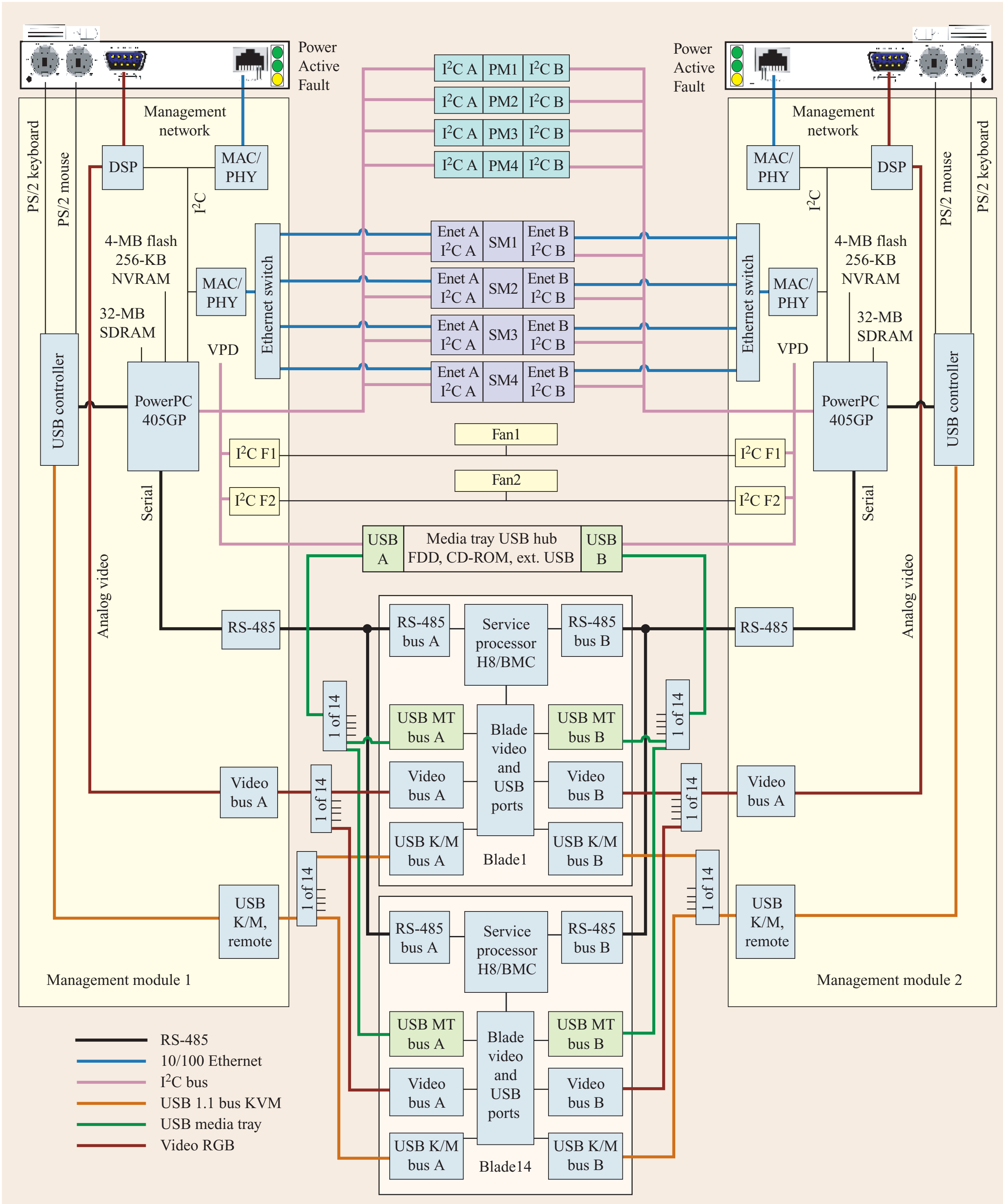

Figure 1

The management module (MM) is an embedded system that provides systems management to the BladeCenter system. It contains a PowerPC processor, ECC memory, and various other interfaces. (DSP: digital signal processor; Enet: Ethernet; PM: power module; SM: switch module.)

WSOU-ARISTA00001299

MMs), they operate in a redundant fashion such that when one component fails, the redundant component takes its place. The midplane is not redundant at the system level, but contains multiple paths to and from each component and is capable of surviving a fault. Redundant components provide continued operation in the event of a failure on an individual component. As shown in Figure 1, only one set of signals is active at any given time. Each component is directed to use one set of signals rather than the other by means of selection signals set by the MM. If one selection signal is active, it indicates that the signals to the first MM should be used. If the other selection signal is active, it indicates that the signals to the second MM should be used. Both signals being active and both being inactive are invalid conditions, and the system components assume that no functional MM is available. In addition to the redundancy offered with duplicating components in the chassis, the systems management hardware infrastructure offers the following redundancy:

- Control signals from each blower.
- I^2C buses from each MM that monitor and control the components installed in the chassis.
- RS-485 buses used for communication between the MMs and the processor blades.
- USB buses that provide a keyboard, mouse, CD-ROM, and floppy disk drive (FDD) to each processor blade.
- Internal Ethernet network interfaces from the MM to each I/O switch module.
- Video buses that route video from the processor blade to the MM.

MM redundancy is possible when a second MM is installed in a BladeCenter chassis. Figure 1 shows the chassis infrastructure with the various redundant management buses that provide the redundant MM with the ability to function as the chassis management module. The primary MM actively monitors and manages the chassis, while the second MM runs in a standby mode. Each MM monitors the state of the other MM. If the primary MM is removed or fails, the second MM signals a failover condition and takes over as the active MM. The failing MM is rebooted to standby status, while the other MM is enabled with all of the configuration settings of the primary module. A failure of the primary MM allows the standby or redundant MM to continue chassis management services to both system administrators on the private management network and to the many hardware modules within the chassis. Since the MMs support hot-plug capability, adding or removing one will not disrupt the current operation or configuration of the chassis components.

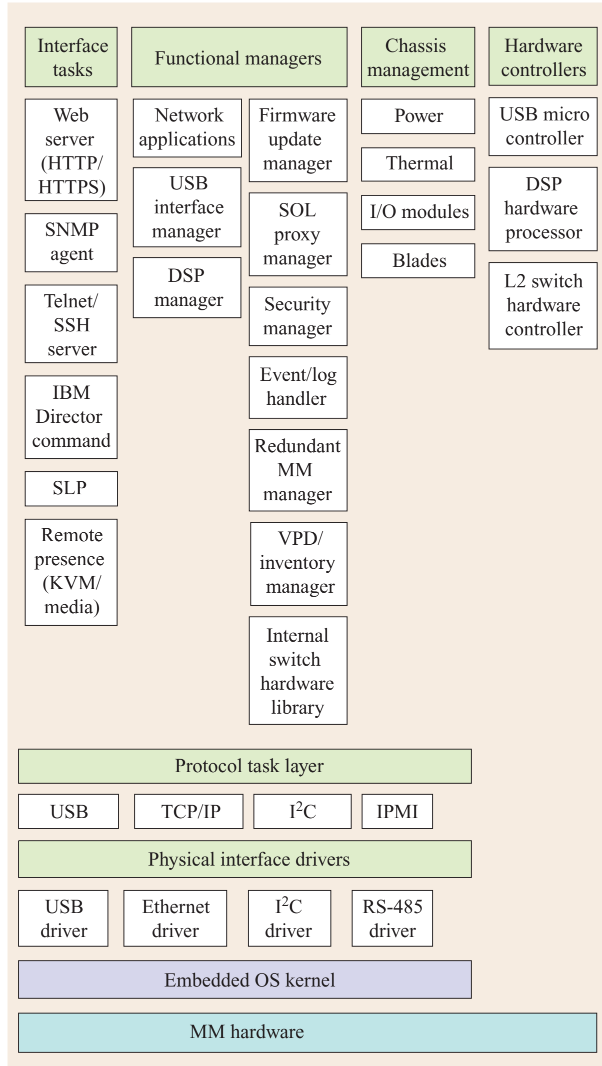

Figure 2

High-level representation of management module firmware.

All components are wired with dual buses to and from the MM. The primary MM has a separate path (bus) to monitor and control devices (blowers, switches, power supplies, the midplane, and the media tray). The standby MM also has unique paths to the same devices. If the MM has problems communicating on the current path, it may failover and communicate via the redundant path. The redundant RS-485 bus (the bus not being used by the primary MM) is used for the dual MMs to communicate with each other. This path is used to keep the standby MM synchronized with the primary MM.

There are two reasons for the primary MM to remain synchronized with the standby MM: MM firmware update and chassis configuration settings. When the

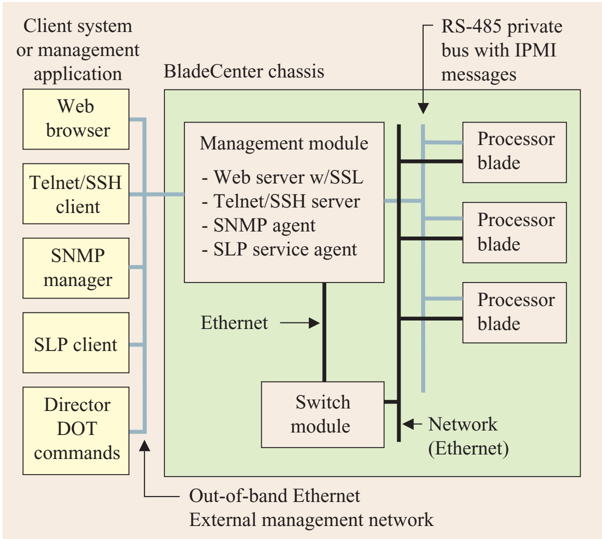

Figure 3

BladeCenter management interfaces.

primary MM firmware is updated through an administrator action, the standby MM firmware is automatically updated without further action from the administrator. This feature was designed so that customers would not have to spend additional time updating the standby MM and to ensure that the MMs maintain a consistent chassis behavior and configuration across failovers. The second reason to stay synchronized is to keep the standby MM in a state in which it can take over for the primary when a failover occurs. During normal operation, when both the primary and standby MM are present in the chassis, checkpoint information is passed to the secondary over the redundant RS-485 bus on an ongoing basis. Checkpoint information includes indications such as lists of active events and faults and the MM chassis event logging that has occurred on the active MM. Having this information checkpointed between the primary and standby MM results in a consistent chassis event log and also prevents duplicate alerts from being sent for conditions that existed before the point at which the takeover occurred. Although the current external user session or connections to the MM (Web browser or Telnet client) will be terminated when a failover occurs, users can reconnect and continue performing systems management functions. After a failover of the MM, the standby MM sends an alert to the system administrator after it becomes the primary MM.

The MM maintains configuration and policy settings of the chassis even when only a single MM is present. In preparation for an MM failure, persistent data is stored in NVRAM on the midplane. The midplane is not a hot-swappable component and is therefore an integral part of the chassis. When an operational MM is inserted into the chassis, persistent data from the midplane NVRAM is used to customize the newly inserted MM with the last known configuration and policy settings of the chassis. The NVRAM on the midplane contains static information, such as user profiles and network settings.

By design, when two MMs are present, the MM in bay 2 has an initial startup delay; this removes the initial power-on race condition and provides a repeatable power-on bias for the MM in bay 1 to become primary. The bias for the MM in bay 2 is to be the standby MM and serve as a backup MM in the event of a failure in the primary. By design, MM hardware failures force the MM to reset. Hardware signals inform the other MM that the redundant MM is in reset. When the primary MM fails or is removed from the chassis, the standby MM detects these conditions and promotes itself to primary. The standby may promote itself only if it detects that the primary is not present or is in reset.

External and remote systems management interfaces

The MM is the central point of control for the configuration and monitoring of all components within a single chassis. Since enterprise customers have many different approaches to systems management, the MM must expose many unique controls and policies so that many disparate customers can incorporate BladeCenter servers into their information technology (IT) complex. Therefore, a number of external interfaces exist to accommodate a large number of customer environments (**Figure 3**).

The MM enables out-of-band management of the chassis and components over an Ethernet network interface. The MM supports four external management interfaces that allow the administrator to monitor, configure, and control the various components within the chassis. The external interfaces provided by the MM are a Web-browser-based graphical user interface (GUI), a command-line interface (CLI) [12], an SNMP agent, and a proprietary command interface called the *DOT command interface*. Management of the behavior of the I/O switch module on the external network is done via Ethernet interfaces into the I/O switch modules independently of the MM.

The MM provides multiple industry-standard protocols with various levels of security corresponding to the interface protocols. The MM offers administrators a robust set of security features to support all components in the chassis without adding additional cost to the component itself. Many IT installations require administration groups or individuals to manage BladeCenter chassis as a whole or in divided subsets at

the component or task level. The MM implements user profiles that can be defined to give each administrator various access rights. By default there is one user profile with supervisor access rights. User profiles can be kept on the MM NVRAM (local) or on a Lightweight Directory Access Protocol (LDAP) server (remote) [13], and users can be authenticated against these local or remote user profiles.

Clients connecting to the MM using a CLI, Web browser, or SNMPv3 must be authenticated. Increased security is provided for Web browsers and LDAP connections by using Secure Sockets Layer (SSL) protocol: SSH is used for CLI, and userids and passwords with data encryption are used for SNMPv3. An added level of security is provided by the MM in that all logins to the MM are logged into the chassis event log (CEL).

The Web-browser-based interface provides a graphical, nonscriptable user interface for access to the MM using an embedded Web server that supports a standard Web browser client using either the Hyper Text Transfer Protocol (HTTP) or the HTTP over SSL (HTTPS) protocol. The CLI is provided by either an embedded Telnet server that supports a Telnet client or an embedded SSH server that supports an SSH client. The SNMP interface is provided by an embedded SNMP agent that supports an SNMPv1 or SNMPv3 browser. The DOT command interface is an IBM Director interface that provides all of the functions provided by a Web user interface except remote control. These commands are used to query and control a BladeCenter chassis for administrators who make use of IBM Director.

The SNMP and CLI provide a programmatic scriptable interface for management automation. A remote client application can therefore use these interfaces for third-party management applications for simple command and response scripts.

The Service Location Protocol (SLP) network service is implemented by the MM SLP service agent (SA). The SA responds to discovery SLP packets for the MM service type and returns attribute information about the MM and chassis. This assists enterprise managers such as Director to automatically locate a processor blade chassis in the network without first determining the Internet Protocol (IP) address of the chassis.

External management interfaces allow clients to query, configure, and control components in the BladeCenter chassis. These include the following:

- *Status:* Health (sensors such as temperatures, voltages, and sensor thresholds); logging [events such as hardware errors, predictive failure analysis (PFA) events, userid log in and out events, and configuration errors]; inventory via vital product data (VPD) (information such as the part number and serial number of each device); LEDs (the MM has the ability to display all front-panel LEDs remotely); and fuel gauge (provides the administrator with the ability to determine the maximum amount of power being consumed by components plugged into the chassis on a power-domain basis).
- *Power management:* The ability to power every pluggable component on and off and display its maximum and minimum power consumption value.
- *Processor blade remote KVM control:* The MM allows the administrator to configure the processor blade connected either locally to the KVM ports on the MM or remotely to a client using the Web browser.
- *Media tray and remote disk:* Using the GUI, the MM can control what is needed to mount network storage for processor blades over USB.
- *Firmware update:* Using the GUI or CLI, the MM can update firmware on the MM and can flash update the BMC on each processor blade.
- *SOL:* Using the Web browser GUI, the MM can set up each processor blade to provide SOL to the MM and set up the MM to provide clients on the private network to access SOL (serial port) data from one or more processor blades. SOL configuration can be performed via any access method, but the SOL function itself is available only via CLI.
- *TCP/IP port numbers:* The MM allows for reconfiguration of port numbers of HTTP, HTTPS, Telnet, SSH, SNMP agent, and SNMP.
- *Management of the SSH server host key.*
- *Ability to identify a processor blade using a blue identify LED.*

Chassis monitoring and error reporting

The MM uses the redundant I^2C buses to monitor the chassis components, and it uses the RS-485 buses to monitor the processor blades and the other MM. Examples of various conditions monitored and reported by the MM are shown in **Table 1**.

Consolidated log

The MM accumulates informational, warning, and error events in a consolidated CEL. This log contains entries generated by various conditions detected by the active and standby MMs and any events or alerts sent to the MMs from the processor blades. Processor blades may maintain their own system error log, but always send their log entries to the MM to be saved in the CEL.

The MM event log (i.e., the CEL) can be viewed, saved, and cleared. An administrator can choose to selectively view the log entries, which can be sorted and filtered by severity, logging source, and logging timestamp in

WSOU-ARISTA00001302

Table 1 Conditions monitored and reported by the management module.

Component	*Conditions*
Chassis	Presence detect (insertion and removal) of chassis components
	Ambient room air temperature entering the chassis
	Communication reliability of the I^2C and RS-485 buses
Power modules	Over-temperature conditions
	Over/under-voltage faults
	Over-current faults
	Failure to share the load with the other power module
	Power boundary capacity
Blowers	Measured speeds using speed control and tachometer interface directly from the blowers
	Blower predictive failure analysis (PFA)
I/O switch modules	Over-temperature conditions
	Over/under-voltage faults
	Power on self test (POST) failures
Voltages	Measured voltage levels excluding blades
Blades	Environmental factors such as voltage, temperature, and various status bits reported by the BMC on the blade
	Boot failures and memory failures reported by POST BIOS or the systems management interrupt (SMI) handler on the blade
Management modules	Over-temperature conditions
	Voltage
	BIST
	Thermal
	The state of the other management module

ascending or descending order. Additionally, the administrator may elect to view or conceal those log entries that are also written to the processor blade error log.

Alerts
When the MM detects a change in the chassis status or receives an alert from the processor blade, it writes an entry describing the event in the CEL. If that event is significant, an alert can be sent to notify a system administrator of the event. These significant events are organized into three alert categories: *critical, warning*, and *system*. Critical alerts are triggered by events such as the following: a power module or multiple blowers fail, a voltage reading goes out of specification, or an invalid configuration is detected. A warning alert is triggered when a noncritical failure event occurs, such as the failure of a single blower or a redundant module. System alerts are triggered by events such as the following: a component is powered on or off, there is an inventory change, or the CEL log is 75% full.

Alerting mechanisms
The user can selectively decide which of the above events should trigger an alert notification to be sent to the administrator. The system can be configured to send alerts to as many as 12 recipients. Each alert recipient can be configured to receive only critical alerts or administrator-selected alerts. Each alert recipient can be configured to receive the alert by one of three methods: an e-mail notification as specified by RFC2821—Simple Mail Transfer Protocol, an SNMP trap as specified by RFC1215, or an IBM Director-formatted alert (a proprietary packet sent over the network that contains structured information describing the alert).

Inventory management and chassis configuration validation
Inventory management comprises the validation, inventory, and tracking provided by the MM when a component is plugged into the chassis or when ac power is applied to the system. The MM also provides a number of chassis configuration checks for all BladeCenter components in the chassis. This checking is run either when the component is plugged into the chassis or when ac power is applied. Two such validation checks are 1) protocol-mismatch checking between the processor blade NICs and I/O switch module and 2) chassis power consumption.

For protocol-mismatch checking, the MM compares NIC protocol types on the processor blade to the port protocol types on the corresponding I/O switch module. If they do not match, power permission is not granted to either the processor blade or the I/O switch module. Power permission is a concept unique to the BladeCenter design. Processor blades and I/O switch modules are prevented from powering on for any reason until they receive permission from the MM. When a protocol reported by the processor blade does not match a protocol reported by the I/O switch module, the MM will issue a critical alert to the system administrator. In addition to protocol checking, the MM checks the power consumption of various chassis components and the oversubscription policy in effect for the power domain before granting power permission to a component.

WSOU-ARISTA00001303

The MM also makes note of the current chassis universal unique identifier (UUID) stored on the midplane VPD device. This information is used for two purposes. First, the chassis UUID is sent to blades and switches within the chassis so that IBM Director (or another systems management entity) may identify the system topology and determine which I/O switch modules and processor blades are contained in which chassis. Second, the chassis UUID and each component UUID are stored in the MM NVRAM. When chassis ac power is removed and then restored, the MM determines whether the UUIDs are the same as before the removal of ac power. If so, the MM powers on all of the switches and blades that were powered on prior to ac power failure. Components that were powered off before the loss of ac power will remain off unless the administrator instructs the blade to power on.

Each electronic field-replaceable unit (FRU) stores VPD in an onboard electrically erasable programmable read-only memory (EEPROM) device. VPD is written as part of a component manufacturing process and can be used as a static repository for device information, such as inventory data. Access to VPD by the MM is a required capability of all components, and the VPD must be available as soon as the component is plugged into the chassis. The fundamental use of VPD is to identify the UUID, part number, and serial number of each component. Additionally, VPD is a convenient repository for receiving information on many aspects of the component, such as the media access controller (MAC) addresses of the NICs, the power consumption of the component or, in the case of the power management, the power output of a power module.

The MM manages component and activity logs with data from VPD devices on each component. The component log contains information uniquely identifying each supported device that was ever present in the chassis. The activity log records the addition and removal of each component listed in the component log. This provides the administrator with a history of components in the chassis as components are moved from one chassis to another. VPD data is made available to management applications external to the MM, which allows them to collect data on multiple chassis and chassis components.

Power management

Power management allows the administrator to set power-management policies for redundant power supplies within a BladeCenter power domain. Power modules that supply power to the chassis must be easy to plug into or unplug from the chassis. The demands of processor blades and other chassis components on power supplies are constantly increasing. To provide that amount of power from a unit the size of a power module requires a balance of power management, cost, package density, and redundancy. When there is a matched redundant pair of power modules (both power modules have the same capacity), the output of both supplies is more than sufficient to provide power to all blades even when running a full load. If one of the power modules fails, the remaining module must be able to provide power to the remaining components in the power domain. Reducing the power consumed by each blade by throttling or slowing down the blade prevents the remaining power module from overheating and shutting down. The term *throttling* is used in this context to represent the ability of a blade to reduce its demand for power.

Power in the chassis is distributed as two separate power domains, with each domain serving a subset of the components in the chassis. Each power domain can contain up to two power modules which share the power load.

Various power modules are offered, with capacities ranging from 1,200 watts to 2,000 watts. Components can be configured in the chassis such that their total maximum power demand exceeds the capacity of a single power module but is within the total capacity of two load-sharing power modules. In this situation, known as *oversubscription*, if one of the power modules should fail, the remaining power module would not be able to fulfill the power demand and would also shut down unless the blades could reduce their demand for power.

It is also possible to configure a chassis with nonmatching power modules, in which case the smaller of the two power modules is used to calculate the output capability of the power domain when a supply fails. The MM firmware provides a number of power-management policy selections that allow the administrator to choose the behavior when the demand in a power domain exceeds the capacity of a single power module. Two of these selections are described below.

The first is *no oversubscription*. New components inserted into the domain are allowed to power on only if they can operate at maximum capacity (without throttling) when power redundancy is lost in this domain. If the demand exceeds the capacity of the smallest power module in the domain, the blade is not allowed to power on. This policy allows the administrator to ensure that components that are powered on will run at full power before and after a power module failure.

The second is a *recoverable oversubscription*. Newly inserted components are allowed to power on only if the component has the ability to throttle down sufficiently to maintain operation when power redundancy is lost in the domain. If the demand exceeds the capacity of the smaller power module in the domain, the MM calculates power-reduction values on the basis of the ability of the blades to

throttle down processors to reduce power consumption. If the total power demand can be reduced below the capacity of a single power module by throttling, the MM calculates the power reduction value for each blade capable of throttling. It sends these values to the blades and then allows the blade to power on. If the total power demand cannot be reduced to the capacity of a single power module, the blade is not allowed to power on.

This policy allows the administrator to run more components at full power when both power modules are functioning properly. Only when a power module fails does the component have to reduce its power consumption.

Thermal management

Blowers are used to cool all of the components in the chassis. In the BladeCenter enterprise chassis, two blowers installed in the rear pull air from the front of the chassis to the back. The air entering the chassis first flows through the blades and then over the I/O switch, power, and management modules. Management of the blowers is implemented with direct tachometer and speed control connections.

The MM controls the speed of the blowers to provide the airflow required to cool the components. The speeds are adjusted by sensing the temperature of the air throughout the chassis. If the temperature of the air is below a minimum threshold, the blowers are set to run at a slow speed. As the air temperature varies, the MM adjusts the blower speed accordingly. When the temperature exceeds a maximum threshold, the blowers are set to full speed. The blowers are also set to run at full speed when various faults occur:

- If a blower fails or is removed, the other blower is set to run at full speed.
- If a power module fails, the remaining power module is the sole source for all power required in its power domain. The blowers are set to full speed to provide additional cooling for the remaining power module.
- If an I/O switch module, a power module, or a blade indicates a thermal warning in which the temperature of the component is approaching a critical temperature, the MM sets the blowers to full speed.

When the warning condition subsides, the MM returns the blower speed to the setting prescribed by the temperature of the air throughout the BladeCenter chassis.

Blower speeds are designed to meet certain acoustic requirements when they run at their slowest speed. If a blade requires additional airflow, it is possible that the acoustic requirements will no longer be met. To address this situation, a concept known as *acoustic mode* can be implemented by the MM if the blades in the chassis have the ability to throttle. When these blades are in acoustic mode, they reduce their power dissipation by throttling down the processors. This reduces the need for additional airflow and allows the blowers to operate at the slowest speed. External interfaces may enable or disable the acoustic mode policy for the chassis. If acoustic mode is enabled by the system administrator, performance is reduced in favor of acoustics. The blowers remain at the slowest speed, and processors on the blade are throttled down until additional airflow is no longer required to cool the blade. If acoustic mode is disabled, processors remain unthrottled, and blower speeds can be increased to provide additional airflow as required.

The blower tachometers are monitored by the MM to determine the blower speeds and to determine whether a PFA and/or fault condition exists for the individual blowers. A PFA event and alert are generated when blower speed drops below 80% of the requested speed. A blower fault event and alert are generated when the blower speed drops below 80% of the minimum speed. Detection of a blower fault results in the MM setting the blower fault LED on the failing blower, a general fault LED on the chassis display panels, and logging the event in the CEL.

MM discovery based on SLP

While it is possible to detect MMs on the network by scanning for SNMP ports or other known ports, the MM supports standards-based network discovery via the Internet standard SLP, which is defined in RFC2165 and RFC2608. This protocol provides a framework to allow networking applications to discover the existence, location, and configuration of networked services. The protocol is designed to simplify the discovery and use of network resources, such as printers, Web servers, mail servers, service processors (i.e., BMC), and other services, such as the MM within the processor blade chassis. The result is that when service instances (such as MM) are added on a network, they are quickly visible to clients, and when they are removed, they are no longer visible. The MM implements an SLP SA and advertises a systems management service to the network. The SA on the MM responds to SLP discovery requests and returns its attributes.

Local and remote KVM, media tray, and remote disk

In standalone servers, a KVM with attached cabling provides a user interface to manage a server in both the preboot and OS environments. Standalone servers also offer access to mass storage devices, such as floppy disks and CD-ROMs typically used during server configuration or while the OS is active. A BladeCenter chassis provides a floppy and CD-ROM located in the MT. The system

removes the need for each server to provide KVM connectors, KVM cables, floppy and CD-ROM drives. Instead, a single KVM, floppy, MT, and USB port is shared across all servers in the chassis. Moving these functions from the server into the chassis reduces the cost, complexity, and installation time of each processor blade.

The BladeCenter design provides the ability for each processor blade to access the local chassis KVM and MT. In addition, the MM also allows for an administrator on the MM Ethernet network to have remote KVM access and the ability to mount remote storage (floppy or CD-ROM) on a processor blade, as shown in **Figure 4**. The processor blade accesses a remote storage device as a local, USB mass-storage device.

The MM contains connectors for a PS/2 keyboard, PS/2 mouse, and SVGA/VGA video used to directly attach KVM locally to a BladeCenter system. Internal to the MM, the PS/2-style keyboard and mouse are converted to USB. The MM controls which one of the 14 processor blades has access to the local KVM and MT. The MM can connect the local chassis KVM to the processor blade or provide a remote KVM over the MM Ethernet port. The BMC in each processor blade directs the KVM or MT USB data to the active MM. The KVM and MT can be assigned to processor blades independently by the administrator. The remote administrator can retain KVM and MT control by disabling local KVM and MT switching via the policy option provided by the external MM interfaces.

Selection of KVM or MT can be changed using any of the following methods:

1. Locally using the push button on a processor blade front panel. When a user presses the front panel button for the KVM or MT selection, the BMC registers the event and sends the user's request to the MM via the RS-485 bus. Depending on the MM policy, which may disable local KVM and MT access, this action may be ignored while the remote administrator controls the chassis.
2. Remotely using the MM external interfaces such as the Web interface. The request to change the KVM and MT is received directly by the MM over the network and the KVM and MT selection is handled.
3. Using a local keyboard hot key sequence, [`NumLock` + `NumLock` + `number (0-9)` + `number (0-9)` + `<enter>`], which provides function similar to a KVM switch for 14 servers, allowing convenient administrator access to any processor blade.

When a different processor blade is selected to access the KVM or MT, the current processor blade using the KVM or MT must be notified to release the resource. The MM informs the current processor blade to release it by sending a message over the RS-485 bus and waiting a predetermined amount of time before the KVM or MT is reassigned to the newly selected processor blade.

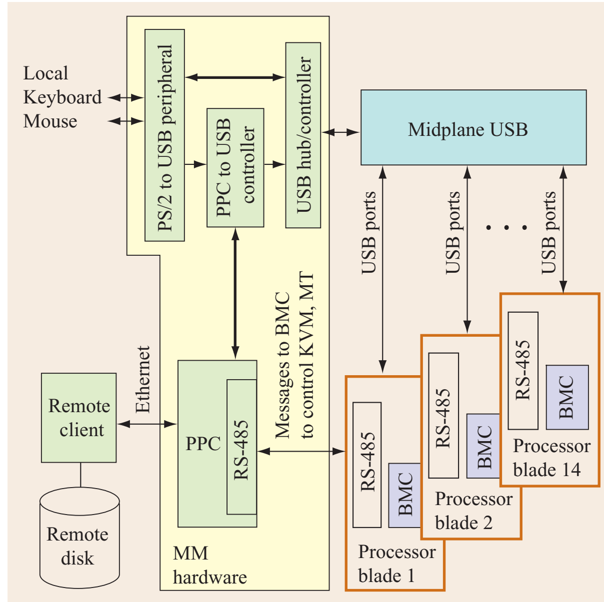

Figure 4

BladeCenter keyboard, mouse, and media tray control.

BladeCenter remote disk feature

Remote systems management often requires access to media for booting processor blades, applying OS patches, or applying firmware updates without physical access to the BladeCenter system. The MM provides a remote disk feature that allows a remote user who has established a Web browser session to the MM to mount one or more of the client drives to a processor blade. Up to 16 supported drives can be connected simultaneously to a single processor blade using the remote disk feature. Drives appear as if they were connected directly to the processor blade as a USB drive.

The remote disk option on the MM Web interface allows the user to mount or unmount up to 16 drives to the current KVM owner. There is also an option to write-protect the drives that will be mounted in order to prevent data from being copied to the mounted drives. Drives are unmounted automatically when the remote control or Web interface session with the MM is terminated, either by the remote user or by the MM.

Often firmware or OS updates must be made to multiple processor blades in the chassis. As a matter of convenience, one diskette drive or drive image can be uploaded to MM memory and used as a local diskette drive by a processor blade. When this feature is enabled, the network connection is needed only for the initial

upload and unmounting of the drive. This allows the disk image to remain mounted on the BladeCenter system even after the Web interface session is terminated. There can be only one drive or diskette image saved on the MM memory at any given time, and its size is limited to 1.44 MB.

Figure 4 shows up to 14 processor blades connected over USB on the midplane to the MM. The MM controls the selection of the local KVM by the MM to one of 14 blades using messages sent on the RS-485 interface.

A disk on the remote client can be mounted by the MM using the Web interface. The MM provides access to the remote disk using the USB ports from each processor blade. Note that the processor blade that is currently assigned the KVM is the only blade permitted to access the remote disk.

Serial over LAN

Serial over LAN (SOL) allows administrators to enable the serial port that exists on the processor blade to transmit and receive character serial data streams to and from a remote Telnet or SSH client connected to the MM external Ethernet port. Since processor blades and the BladeCenter chassis have no external serial port connectors, SOL is the only mechanism provided to allow for interaction with the processor blade serial port.

Some IT installations require high levels of automation that necessitate remote management on a scriptable text-based OS interface. SOL provides an excellent solution for clients to remotely manage a text-based OS. Additionally, some processor blade designs may not wish to have the added expense of a video chip on the blade, since their target customer is satisfied with only text-based interaction.

The MM SOL proxy support allows for an aggregation point for all of the serial data traffic that flows between external clients on the MM Ethernet network and all processor blades in the chassis. The MM providing the role of an SOL proxy service will establish an IPMI SOL session to the processor blades [14]. Thus, 14 concurrent Telnet/SSH sessions can be established to the MM, which in turn will allow for interaction with each processor blade.

SOL support represents a client establishing a Telnet and/or an SSH session to the MM over the MM external Ethernet port. SOL support is a required function that must be implemented on all processor blades.

BladeCenter video

The MM offers a video solution for processor blades that allows either local or remote monitoring of video for one of 14 blades. The local monitoring uses a video graphics array (VGA) monitor connected to the MM. Remote monitoring is provided via a network connection to the MM using a Sun Java**-enabled Web browser.

Remote video support allows the administrator to continually monitor a processor blade, including monitoring or debugging a blade if it fails during POST. In addition, after the OS is loaded, both text and graphics modes are supported. Remotely accessing a processor blade video stream over Ethernet allows a user to maintain a video quality of at least 1,024 × 768 pixels and a minimum five-frame-per-second update rate. Compression is used to minimize network traffic and increase perceived performance.

BladeCenter video hardware spans the MM, midplane, and processor blades. In conjunction with the BMC in the processor blade, the MM controls which blade is selected. The midplane switches the selected processor blade video signals to the MM for local and remote viewing.

Blade server BMC overview

A BMC on each processor blade provides the MM with a common interface to manage various processor blades. The MM communicates with the BMC in each blade to support features such as power on or off requests, error and event reporting, KVM requests, and requests to use the shared media tray over the RS-485 bus. In addition to supporting the MM RS-485 interface, the BMC must also provide the systems management function within the processor blade itself.

Figure 5 shows how the BMC operates in BladeCenter systems. The BMC is an embedded microcontroller dedicated to systems management covering all aspects of the processor blade. Systems management encompasses blade in-band, out-of-band, and side-band messages to and from the BMC. In-band messages are messages between the BMC and the host OS; out-of-band messages are messages between the MM and the BMC; and side-band messages are controls within the processor blade, such as power controls and systems health monitoring and reporting. They also work in conjunction with the MM to control shared chassis resources such as KVM, media tray, remote disk, power and thermal management, FRU inventory, and SOL.

The Renesas Technology H8STM microcontroller is used as the blade BMC. The power of this microcontroller lies in its feature-rich set of memory-mapped peripherals, which include analog-to-digital converter, serial ports, I^2C bus interface, low-pin-count (LPC) Peripheral Component Interface (PCI), pulse-width modulation controllers, general-purpose I/O, timers, and data transfer controller. The analog-to-digital converter is used to monitor voltage levels and current loads.

The serial ports of the service processor and BMC have several uses. The interface to the MM is a serial port using RS-485 transceivers and a custom protocol. Redundant RS-485 buses and selection are done entirely

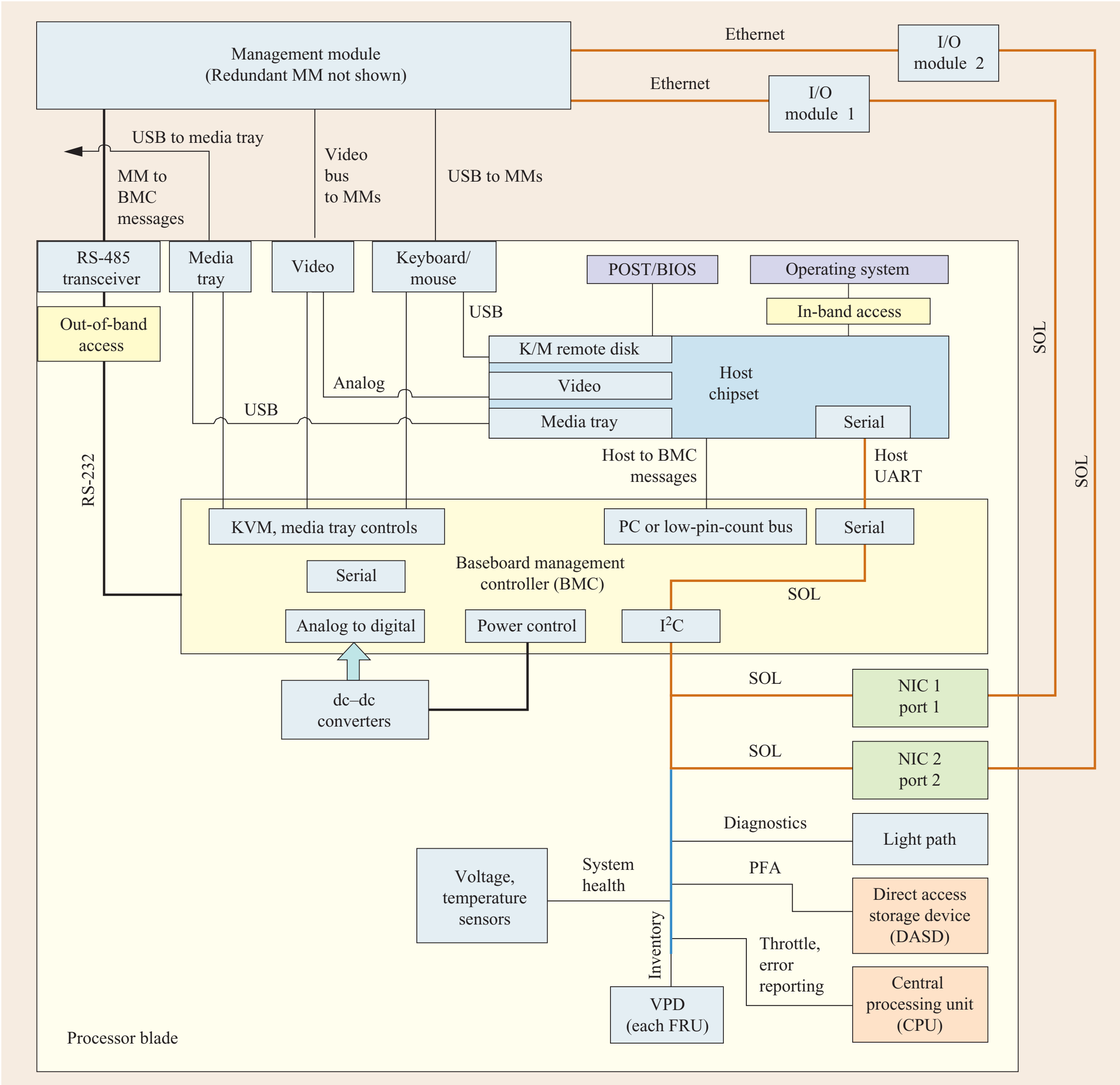

Figure 5

Role of the baseboard management controller in the BladeCenter system.

in hardware and are transparent to the microcontroller. The RS-485 bus is a multidrop topology and thus requires collision avoidance and detection algorithms. A second service processor serial port is used to capture and redirect serial traffic over the processor blade LAN interface, i.e., SOL.

The BMC, in conjunction with sensors located throughout the processor blade, monitors subsystems that detect hardware failures, predictive failures, and status, such as the voltage or current values. Some server designs have dozens of I^2C bus devices connected to the BMC. This may include temperature and voltage sensors, digital I/O expansion, and multiplexers to connect additional I^2C buses. FRUs also store VPD in I^2C bus memory devices that can be retrieved via the service processor.

The BMC contains several software 8-bit and 16-bit timers. Timers are used to provide periodic task dispatching and the BMC firmware watchdog, which is a timer that detects a BMC firmware failure. The watchdog timer must be periodically refreshed by BMC firmware; if the firmware fails to refresh the timer before the timer expires, the BMC is rebooted.

The BMC firmware provides basic control of the processor blade, such as local and remote power on, power off, and system reboot. BMC firmware can also detect failed host central processing units (CPUs). When a host CPU has failed, it is disabled, and the server is typically rebooted with remaining CPUs. In addition, the BMC can detect whether the POST/BIOS firmware running on the host has failed by using a watchdog timer that is armed upon power on and reset by POST/BIOS firmware. When the BMC detects a failure reported by the watchdog timer, the BMC selects the backup POST/BIOS image and the system is rebooted.

The main role of the BMC during normal operation is to monitor and report system health and allow reconfiguration of chassis resources. This includes monitoring the temperature of CPUs and onboard dc–dc regulators. Certain CPU and other faults can be detected by the BMC. Once faults are detected, the BMC alerts the MM. To assist in determining failed blades or FRUs, the BMC provides a light-path diagnostic indication to the failed component by illuminating LEDs after the processor blade is removed from the chassis. If a hardware failure on a blade is detected, the fault LED located on the front panel of each processor blade is turned on. Once the failing blade has been removed from the chassis for service, the LEDs on the blade can be illuminated without ac power. A button press on the system board uses a capacitor to reenergize the light path independently of the BMC. Light-path diagnostics will then identify the location of the fault, whether it is a failed system board, CPU, or another FRU in the processor blade.

Another feature of the BMC is the ability to upgrade the BMC firmware in the field. This can be accomplished by one of two methods: in-band (using the OS) or out-of-band (using the MM and BMC). In-band BMC firmware updates require the user to execute a BMC firmware update utility. The out-of-band method via the MM uses one of the external interfaces, such as a Web browser.

Blade BIOS

The BIOS is code that is executed on the host processor when the processor blade is powered on or reset. BIOS code is located in nonvolatile memory on the processor blade and does not require disk access. This code is used to initialize hardware in the blade in preparation for loading the OS (DOS, Microsoft Windows**, Linux**, etc.) from a device on the processor blade, in the media tray, a remote disk provided by the MM, or a network-based program (e.g., PXE [15], BOOTP [16], or iSCSI), which in turn may load the OS over the network instead of from a local device.

The BIOS runs POST hardware diagnostic routines to ensure that each subsystem (CPU, memory, flash, etc.) is running properly. In addition, it enables various watchdogs and works in conjunction with the BMC to give the processor blade the ability to recover from a corrupt BIOS or a hang condition during POST. If failures are detected, they are reported to the MM and logged for analysis.

Faulty hardware is often detected during power-on-reset before the OS is loaded. Servicing the machine requires interacting with the processor blade via the KVM. Provision is made for IT environments where there is no expert available to diagnose these hardware faults or configuration problems. The MM, BMC, and BIOS act in concert to provide SOL capability. For SOL, the BIOS enables the redirection of the COM2 serial port data to the BMC, which in turn passes this data to the MM SOL proxy function. The COM2 port is dedicated to BMC, and there is no external connector.

The system provides a number of different boot options. A processor blade can boot from a local disk or a network; it can also boot from the floppy disk or CD-ROM on the chassis media tray or from a remote disk provided by the MM. Using data set into VPD by the MM, BIOS selects the boot device after POST is complete. In the case in which a boot device does not load an OS, the administrator can set an ordered list of boot devices from which the BIOS will attempt to boot.

Network booting is optional for processor blades. If network booting is selected for a processor blade, the NIC vendor provides code that is executed instead of loading from a storage device. PXE-enabled NICs enabled for booting will attempt to find a PXE or BOOTP server to load an OS. If the PXE server does not respond, PXE code running on the host will attempt to find the server on another port. This makes use of redundant NIC ports on the processor blade to account for situations such as failed NICs or switches or an improperly configured network. Typically, PXE code is run on all NICs to provide failover to account for situations such as failed NICs or switches or an improperly configured network. In addition, the user may select a network boot option via a BIOS setup menu on a particular NIC.

After the OS is operational, BIOS interacts with the BMC to monitor and respond to various errors, such as memory faults and power management, detected within the processor blade. Operating systems are informed of the capabilities of server hardware by passing information

in memory to the OS using systems management BIOS and Advanced Configuration and Power Interface tables. To facilitate managing BIOS flash updates, the BIOS reports firmware levels in VPD, allowing the MM to query this data at any time and providing external access to this information. Ethernet MAC addresses, Fibre Channel World Wide Name (WWN), and InfiniBand** Globally Unique Identifiers (GUIDs) are collected by BIOS and placed in VPD, informing the MM and other external systems management software applications of physical addresses of the NICs for network configuration and troubleshooting.

Manufacturing data—such as blade serial number, machine type or model number, UUID, and asset tags—is necessary to uniquely identify a processor blade. Manufacturing systems provide a unique serial number across a given machine type or model. The serial number and machine type or model information can be used by IBM service personnel to identify a particular family of blades before providing help-center support or to determine warranty information on a particular processor blade. The UUID is a 16-byte value used to uniquely identify a processor blade. Enterprise management software, such as IBM Director, uses this value to uniquely identify a component within a chassis. Asset tags are customer-assignable values and are opaque to BIOS.

As the BIOS initializes hardware subsystems and runs POST, the BIOS reports status through the standard video port or through the serial port when SOL is enabled. Additionally, the BIOS logs information into the MM CEL to log such items as boot status (checkpoints), PFA, hardware, and boot failures, providing a unified view of all server boot attempts and failures in the chassis.

Blade power and thermal management

BIOS plays an important role in managing power in the BladeCenter chassis. All processor blades have the ability to detect the presence of optional components in the processor blade without powering on the blade, but current high-volume industry-standard chipsets do not have the ability to detect the power consumption of the optional components. For example, the presence of a CPU can be detected by the BMC before power on, but the amount of power the CPU will consume when it is powered on can be determined only by powering on the processor blade and allowing BIOS to query the CPU type to determine its maximum power consumption. After power from power domain 2 is applied to the processor blade, BIOS informs the BMC of the maximum amount of power used by the CPUs of the processor blade. The BIOS then waits for a signal from the BMC to proceed to boot the processor blade. If the MM determines that the power domain cannot supply the power needed to run the processor blade, the BMC is informed to turn off power to the blade.

A condition can exist whereby a power-module failure reduces the available power to a power domain in the chassis. Some CPU implementations require that a hardware register in the CPU be changed to reduce the power consumption of the CPU. The BMC signals the BIOS through a systems management interrupt (SMI) that a power module has failed, and the BIOS then sets the CPU hardware facility to reduce the power consumption of each CPU in the blade. In addition, the BIOS returns the CPU to full power when the MM, through the BMC, reports that the power module is not in a failed state. In addition to controlling the power consumption on a power domain basis, the BIOS reduces the processor blade power consumption when the CPU has reached a thermal threshold, a chassis blower fails, or due to the chassis acoustic mode policy.

I/O switch module management

This section describes how the MM and the control point (CPU subsystem) in an I/O switch module work together to provide systems management in a BladeCenter chassis. The I/O switch module provides connectivity over a midplane between the processor blades and the external switch interface uplink ports. Two types of communication paths exist between the MM and the switch module. The first is a low-level interface implemented as a standard I^2C bus, and the second is a higher-level communication link implemented as an Ethernet interface.

When an I/O switch module is plugged into the chassis, power domain 1 immediately receives standby power. The MM uses I^2C bus registers and VPD to communicate with the switch module. The MM queries the switch module for identifying information and then determines whether the chassis configuration (i.e., protocol verification between the blade NIC and the switch module) is a valid one. If the configuration is valid, the MM turns on the power (power domain 2) to the switch module to perform initialization. Once powered on, the switch module runs POST diagnostics and reports failures to the MM. If POST completes successfully, the switch module internal MM Ethernet interface is enabled.

All switch modules support management external to the BladeCenter chassis and via the public network by implementing an SNMP agent to support an SNMP manager, a Telnet, and/or an SSH server to support a CLI, and an HTTP server to support a Web browser interface. In addition, switch modules may support a proprietary management interface that runs over the switch module internal MM Ethernet network or from an external switch module port.

For management purposes, the control point in the switch module is accessible by either the private management network or the public management network. The private management network is defined as the network to which the MM is physically connected via its RJ45 port. The public management network is the network connected via the switch module external ports. Note that by setting MM policy, the administrator determines whether the control point in the switch module can be accessed via a connection over the switch module external uplink ports.

To allow management applications to access the switch module control point from the private management network (and without consuming an additional switch port), the MM exposes the IP address of the switch module control point on the private network. All traffic to the MM from the private network is forwarded to the switch module over the internal MM-to-switch-module Ethernet connection on the midplane. Conversely, all traffic from the switch module to the MM destined for the private network is also forwarded by the MM.

An I^2C bus interface is used by the MM to internally provide control of the I/O switch module and to collect system status and VPD information. The following control and data areas are accessible by the MM on the I^2C bus interface: the VPD EEPROM, Control Register, Extended Control Register, Status Register and Extended Status Register, and Diagnostic Register.

Firmware updates

The BladeCenter chassis has many CPUs spread throughout the chassis. Each unit requires NVRAM to store firmware across power cycles. MMs, BMCs, and some I/O switch modules require periodic firmware updates to provide additional features and fix support. Firmware updates can be initiated through the MM using the Web browser interface, the CLI along with a Telnet/TFTP (Trivial File Transfer Protocol) server, or the MM Dot Command external interface. Regardless of the interface used, a system administrator selects the target component for the firmware update (MM, BMC, or applicable I/O switch module), selects the file on the local system containing the desired firmware image, and then initiates the update.

The firmware images that can be updated for the MM include the main MM application image, the remote graphics image used by Web browser clients, and the MM boot image. The main application image contains firmware for each of the programmable USB devices. When the main application image is booted, the MM detects the version of the images running on the USB devices and performs a background update of the devices if one is needed.

The firmware update for the BMC requires the administrator to select one of up to 14 target processor blades. If the transferred image file passes the integrity and applicability checks, the MM transmits the image over the RS-485 interface, allowing the BMC to write its own NVRAM.

To ensure that a BladeCenter configuration containing two MMs maintains the exact same flash image, the primary MM detects the image version running on the redundant MM. If the redundant MM is not running the same image version as the primary MM, the primary MM automatically flashes its version of firmware onto the redundant MM. This is done as a background operation. This operation may be initiated either when the primary MM is flashed and reset, or if a second MM (the redundant MM) is plugged into the chassis.

Remote disk support can be used by the system administrator to allow firmware updates targeted to BIOS or diagnostic firmware within the processor blade. The MM allows a floppy or CD-ROM on a remote client to be mounted to the processor blade. BIOS can then be configured to boot from the remote floppy or CD-ROM that contains the firmware update package.

Chassis component diagnostics

New and enhanced component diagnostics in the form of problem determination (PD) tools have been developed for customers and service personnel supporting the BladeCenter system. Component diagnostics isolate hardware problems to a FRU or customer-replaceable unit (CRU) for replacement. High-volume industry-standard xSeries* PD tools fall into the four following categories, all of which run on the host system and CPUs being diagnosed.

POST/BIOS: POST/BIOS is an integral part of processor blade hardware and is executed when the processor blade is powered on. POST/BIOS includes BIST and other self-test diagnostics that run on subsystems such as the CPU or PCI devices. Errors detected are reported to the MM over the RS-485 bus and are logged into the CEL. Diagnostic routines running on the IBM Director server can be configured to gather data periodically from the MM CEL, preventing the loss of this data if the MM CEL should wrap because of a full log condition.

During runtime, processor blade BIOS and the BMC isolate errors between FRUs in the blade to minimize the number of FRUs to be replaced when service is provided. For example, when PCI or PCI-Express** errors, such as system errors (SERRs) or parity errors (PERRs), are detected by a bridge and/or reported by a PCI adapter card, it may not be clear whether a PCI card, a host PCI device, or the CPU has failed. The BIOS, along with the BMC, helps to ensure that the CPU is not replaced when a timeout on the PCI bus is detected. A timeout on the

WSOU-ARISTA00001311

PCI bus is signaled as an internal CPU processor error (IERR) by Intel CPUs as they monitor the PCI bus. A failed transaction due to infinite retry on the PCI bus is also reported as IERR by the CPU. Error information is collected from the PCI devices on the bus before a reset is issued.

When an IERR is signaled by the CPU, the blade enters the CPU shutdown state and code execution halts. In this case, interaction between the BIOS and BMC allows the BMC to restart the CPU by issuing a reset, allowing the BIOS to analyze CPU machine data which is saved across the reset by the CPU. Data collected before and after the reset is used to isolate the failing FRU. In general, BIOS will log runtime recoverable and unrecoverable machine check errors from the CPU into the MM CEL, which is useful for failure analysis.

For I/O switch modules, the CPU within the switch executes a proprietary POST and places the results in the diagnostic registers accessible by the MM over I^2C bus. Depending on chassis policy, a number of levels of POST can be run on an I/O switch module spanning the completeness of the diagnostic coverage (i.e., standard, extended, and full) and the length of time (from seconds to minutes) the diagnostic will run.

*PC-Doctor** and IBM DOS ROM-based diagnostics:* This DOS-based ROM diagnostic is also an integral part of processor blade hardware and is executed when the administrator manually runs a hardware preboot test by hitting the key sequence for *Press F2 for Diags* during boot. This diagnostic provides blade-level information and diagnostic tools to isolate problem FRUs and CRUs, such as hard disk drive, memory, CPU, integrated systems management processor (i.e., BMC), network chipsets, and other subsystems.

IBM real-time diagnostics (RTD) [17]: This diagnostic executes after POST/BIOS in application space after the OS is loaded.

IBM RTD BladeCenter diagnostics: Running in IBM Director.

RTD uses the common diagnostic model (CDM) [18], which is an architecture and methodology for exposing system diagnostic instrumentation through the common information model (CIM) [19] standard interfaces. CIM is an extensible object-oriented scheme for system management being developed by the Distributed Management Task Force [20], and is evolving industry-wide as the basis for systems management architectures. The CDM architecture allows vendors to add modules, called *providers*, to expose their subsystem diagnostic expertise using RTD in a pluggable, extensible fashion when running on the host CPU. The RTD application supports all eServer xSeries blades.

Integrating multiple components, such as processor blades, I/O switch modules, MM, and power modules into a single chassis allows diagnostic packages to provide innovative methods to diagnose problems that span multiple components, each running separate diagnostics. Diagnosing all components requires that systems management components—IBM Director Server, the MM, the integrated systems management processor (ISMP), BIOS, the I/O switch module, and RTD—perform diagnostic tasks on the BladeCenter system as a single unit. Packaged with RTD running on IBM Director Server are several IBM Director tasks that support diagnostics and information gathering for all components (processor blades, I/O switch modules, etc.). These tasks provide post-processing of all diagnostic information collected from each component or subsystem and result in service recommendations for all major electronic component FRUs and CRUs monitored by systems management hardware.

The MM provides all external networking communication necessary for a diagnostic such as RTD to run anywhere in the network, for example in Director Server. RTD uses the Director framework to provide a vertically integrated approach to diagnostics from which data can be collected from many components run at various points in time. For example, as mentioned above, POST and BIST run on all components whose results are collected by applications such as RTD. In addition, RTD collects chassis-wide light-path diagnostic status for all components in the system, performs analysis, and describes actions to be taken by service personnel, such as the replacement of a component marked by an illuminated light-path LED.

The MM provides a number of important features for diagnostics. In addition to providing external access to the BladeCenter system for RTD, the MM detects and reports failures and places them in the MM CEL log. The CEL is an aggregated log that timestamps and collects any failure detected in the chassis. MM-detected failures that are placed in the CEL include cases in which a processor blade or I/O SN is present but the RS-485 bus or I^2C bus is not operational, or an I/O switch module reports a diagnostic error (critical or noncritical) after POST has completed.

RTD also controls and/or uses the I/O switch module POST diagnostics. When the administrator desires a diagnostic to be run on an I/O switch module, the RTD task recycles power to the I/O switch module to cause it to run POST in one of the three levels (standard, extended, or full) as described earlier. RTD collects the new I/O switch module diagnostic information, but causes the I/O switch module to drop all network connections for the duration of the diagnostic.

The features discussed above are exposed in the BladeCenter diagnostic RTD task extensions in Director,

WSOU-ARISTA00001312

which appear to the end-user under BladeCenter assistant and are categorized as follows:

- Light-path diagnostics.
- Self-test results.
  - I/O module POST.
  - Blade server: Gathers the self-test results of the BMC or ISMP.
  - MM BIST.
  - I^2C bus test results: Indicate the status of the I^2C bus communication of the MM and blade server management hardware.
- Midplane connectivity: Indicates whether communication through the midplane RS-485 network is operational.
- System event log: Gathers the aggregated chassis system event log (the CEL).
- Restart I/O module extended diagnostics.

RTD makes use of the Director event action plan infrastructure to respond to failures reported by BladeCenter alerts by collecting a snapshot of the above RTD task extensions. Administrators may schedule any RTD BladeCenter tasks to run at low utilization periods in order to minimize downtime should a component require maintenance actions.

The following example of a dual inline memory module (DIMM) error and failure illustrates how the four PD tool types monitor problems by integrating firmware components from BIOS, ROMDOS diagnostics, BMC, and the MM. Suppose that a blade with two DIMMs is subject to a multibit memory error in a single DIMM:

1. *BIOS:* The server hardware generates a nonmaskable interrupt (NMI), and BIOS reboots the server (by default). In addition to an onscreen NMI message, BIOS and the ISMP send an alert to the MM, which generates an entry in the CEL.
2. *ROMDOS memory diagnostics:* If further DIMM testing is desired, the full memory diagnostics log another DIMM failure message (i.e., if the read and write pattern tests fail) and send the resultant standard error code through BIOS to the BMC, which routes it to the MM CEL. Note that under certain circumstances, a single DIMM failure may be such that the processor blade can operate using one of the two DIMMs by disabling the failing DIMM. BMC will then illuminate the light-path LED beside the DIMM, indicating the location of the fault, which is observable from numbers 3 and 4, below.
3. *RTD running on the blade OS:* Will report the memory problem through light-path analysis.
4. *RTD on the Director Server:* Will report the memory problem either by light-path analysis or collection of the MM CEL, which can be automatically gathered using the Director event action plan in response to the MM alert issued to the RTD Director Server in the event of a BladeCenter hardware fault.

In the derivative DIMM failure case, if all DIMMs have failed as detected by POST BIOS or the BMC, ROMDOS cannot be loaded into memory to find the fault, and host-OS-based tools cannot be used, since the OS cannot be loaded without usable memory. In this case, number 4 above remains viable to automate and aid problem determination by administrators and service personnel.

Exploiting error-detection mechanisms in hardware and PD tools (such as RTD in Director, RTD running on the OS, ROMDOS, and the POST/BIOS collection of failures such as DIMM error) results in excellent FRU/CRU fault isolation, requiring little if any interaction by the administrator to analyze the log in order to determine the failing FRU.

Conclusion

BladeCenter chassis management provides a new paradigm for the administration of a number of compute, network, and storage nodes along with environmental components, such as power and cooling modules. Effectively managing the collection of redundant hardware, each with its own embedded management controller, from a single entity is critically important. This allows administrators to quickly install, configure, inventory, and diagnose their equipment from any point on the network. The BladeCenter management module offers numerous interfaces by which an administrator can manage a chassis, allowing the system to fit into any IT complex using common interfaces such as a Web browser, a command-line interface, or an SNMP manager. The systems management provided by the BladeCenter management module requires all components to participate in chassis management and results in a single management scheme coordinated by the management module at the chassis level.

Remote management is a key attribute in today's IT environments. The management module provides the ability for the processor blades to export their keyboard, video, mouse, and serial port interfaces to another node in the network. BladeCenter I/O switch modules are managed over the IP network, allowing switch vendors to provide both standard and proprietary switch management. Shared components, such as power, blowers, and the media tray, are managed at the chassis level via the Web or SNMP interfaces. In addition, shared components are managed at the chassis level instead

of the individual node level, which simplifies the policy settings across groups of components and allows chassis resources to be managed over a group of compute and network nodes.

Acknowledgments

The authors gratefully acknowledge the constructive comments of both Dr. Tom Bradicich of the IBM Systems and Technology Group and Dr. Richard E. Harper of the IBM Research Division, Thomas J. Watson Research Center.

*Trademark or registered trademark of International Business Machines Corporation.

**Trademark or registered trademark of Sun Microsystems, Inc., Microsoft Corporation, Linus Torvalds, InfiniBand Trade Association, or PCI-SIG Corporation in the United States, other countries, or both.

References

1. D. M. Desai, T. M. Bradicich, D. Champion, W. G. Holland, and B. M. Kreuz, "BladeCenter System Overview," *IBM J. Res. & Dev.* **49**, No. 6, 809–821 (2005, this issue).
2. Alert Standard Format (ASF) Specification; see *http://www.dmtf.org/standards/asf/*.
3. IBM Remote Supervisor Adapter User's Guide Version 6.0—Servers; see *www.ibm.com/pc/support/site.wss/MIGR-4TZQAK.html*.
4. IBM Remote Supervisor Adapter II SlimLine and Remote Supervisor Adapter II User's Guide—Servers; see *http://www-1.ibm.com/support/docview.wss?uid=psg1MIGR-57091*.
5. G. Pruett, A. Abbondanzio, J. Bielski, T. D. Fadale, A. E. Merkin, Z. Rafalovich, L. A. Riedle, and J. W. Simpson, "BladeCenter Systems Management Software," *IBM J. Res. & Dev.* **49**, No. 6, 963–975 (2005, this issue).
6. J. Case, M. Fedor, M. Schoffstall, and J. Davin, "RFC 1157—Simple Network Management Protocol (SNMP)," Network Working Group, May 1990; see *http://www.faqs.org/rfcs/rfc1157.html*.
7. M. J. Crippen, R. K. Alo, D. Champion, R. M. Clemo, C. M. Grosser, N. J. Gruendler, M. S. Mansuria, J. A. Matteson, M. S. Miller, and B. A. Trumbo, "BladeCenter Packaging, Power, and Cooling," *IBM J. Res. & Dev.* **49**, No. 6, 887–904 (2005, this issue).
8. Koninklijke Philips Electronics N.V., I^2C, a Two-Wire Serial Bus Communication Standard; see *http://www.philipslogic.com/i2c*.
9. J. E. Hughes, M. L. Scollard, R. Land, J. Parsonese, C. C. West, V. A. Stankevich, C. L. Purrington, D. Q. Hoang, G. R. Shippy, M. L. Loeb, M. W. Williams, B. A. Smith, and D. M. Desai, "BladeCenter Processor Blades, I/O Expansion Adapters, and Units," *IBM J. Res. & Dev.* **49**, No. 6, 837–859 (2005, this issue).
10. S. W. Hunter, N. C. Strole, D. W. Cosby, and D. M. Green, "BladeCenter Networking," *IBM J. Res. & Dev.* **49**, No. 6, 905–919 (2005, this issue).
11. J. E. Hughes, P. S. Patel, I. R. Zapata, T. D. Pahel, Jr., J. P. Wong, D. M. Desai, and B. D. Herrman, "BladeCenter Midplane and Media Interface Card," *IBM J. Res. & Dev.* **49**, No. 6, 823–836 (2005, this issue).
12. IBM Corporation, *Management Module Command Line Interface Reference Guide*, Second Edition, IBM eServer BladeCenter, June 25, 2004; see *http://www-1.ibm.com/support/docview.wss?uid=psg1MIGR-54667*.
13. IBM Corporation, *Lightweight Directory Access Protocol User's Guide—IBM eServer BladeCenter HS 20 Management Module and IBM Remote Supervisor Adapters*, First Edition, IBM eServer BladeCenter, April 2004; see *http://www-1.ibm.com/support/docview.wss?uid=psg1MIGR-55014*.
14. Intel Corporation, Intelligent Platform Management Interface; see *http://www.intel.com/design/servers/ipmi/*.
15. Intel Corporation, Preboot Execution Environment (PXE) Specification, Version 2.1, September 20, 1999; see *ftp://download.intel.com/labs/manage/wfm/download/pxespec.pdf*.
16. B. Croft and J. Gilmore, "RFC 951—Bootstrap Protocol (BOOTP)," Network Working Group, The Internet Society, September 1985; see *http://www.rfc-editor.org/rfc/rfc951.txt*.
17. IBM Corporation, IBM Real Time Diagnostics; see *http://www.ibm.com/servers/eserver/xseries/systems_management/sys_migration/rtd.html*.
18. Intel Corporation, Common Diagnostic Model (CDM); see *http://www.intel.com/design/servers/CDM/index.htm*.
19. Distributed Management Task Force, Inc., Common Information Model (CIM); see *http://www.dmtf.org/standards/cim*.
20. Distributed Management Task Force (DMTF); see *http://www.dmtf.org*.

Received December 16, 2004; accepted for publication February 21, 2005; Internet publication October 7, 2005

Thomas Brey *IBM Systems and Technology Group, 3039 Cornwallis Road, Research Triangle Park, North Carolina 27709 (tbrey@us.ibm.com)*. Mr. Brey is a Senior Technical Staff Member and has worked for IBM for more than 25 years. He joined IBM after receiving a B.S. degree in electrical engineering from the University of Hartford in 1979. Most of his career at IBM has been spent in systems management in S/390* and xSeries systems. Mr. Brey is currently working on the architecture and design of BladeCenter products.

Brian E. Bigelow *IBM Systems and Technology Group, 3039 Cornwallis Road, Research Triangle Park, North Carolina 27709 (bbigelow@us.ibm.com)*. Mr. Bigelow received his B.S. degree in engineering from Kansas State University in 1983. Before joining IBM, he worked in the telecommunications and military equipment industry as a circuit designer with emphasis on communication and system control products. Mr. Bigelow joined IBM in 2001 as an Advisory Engineer and has focused on the development of the BladeCenter chassis management module.

Joseph E. Bolan *IBM Systems and Technology Group, 3039 Cornwallis Road, Research Triangle Park, North Carolina 27709 (bolan@us.ibm.com)*. Mr. Bolan is a Senior Technical Staff Member in BladeCenter development. He joined IBM after receiving a B.S. degree in electrical engineering from Rensselaer Polytechnic Institute in 1979. During his career at IBM he has worked on several different products, including S/390 I/O subsystems, AS/400* I/O processor development, and xSeries systems management. Mr. Bolan currently works on the BladeCenter architecture and design.

Harry Cheselka *IBM Systems and Technology Group, 3039 Cornwallis Road, Research Triangle Park, North Carolina 27709 (cheselka@us.ibm.com)*. Mr. Cheselka is a Senior Engineer. He has worked for IBM for more than 30 years. He received a B.S. degree in electrical engineering from the New Jersey Institute of Technology and an M.S. degree in computer information science from Syracuse University. His career at IBM has largely been spent working on development of various products, including the 3270 subsystem and networking products; he is currently working on requirements for BladeCenter modules.

Zeynep Dayar *IBM Systems and Technology Group, 3039 Cornwallis Road, Research Triangle Park, North Carolina 27709 (zeynepd@us.ibm.com)*. Ms. Dayar is a software engineer in systems management firmware development. She has a B.S. degree in computer engineering and an M.S. degree in computer science. She has been involved with the design and implementation of external user interfaces for the IBM xSeries systems management hardware products. Ms. Dayar is currently working on user interfaces for the BladeCenter management module.

Jeffery M. Franke *IBM Systems and Technology Group, 3039 Cornwallis Road, Research Triangle Park, North Carolina 27709 (jfranke@us.ibm.com)*. Mr. Franke has worked in systems management at IBM for four years. He received an M.S. degree in applied mathematics from the University of Minnesota in 1989. Before joining IBM, he developed software for aircraft landing systems and navigational aids, global positioning systems, and satellite control systems. Mr. Franke is currently working on the software architecture and design of BladeCenter products.

Donald E. Johnson *IBM Systems and Technology Group, 3039 Cornwallis Road, Research Triangle Park, North Carolina 27709 (1johnson@us.ibm.com)*. Mr. Johnson has worked in systems management at IBM for four years. Before joining IBM, he worked at Raytheon Missile Systems developing weapons systems. He has a B.S. degree in electrical engineering. Mr. Johnson is currently working on the BladeCenter management module firmware.

Rajiv N. Kantesaria *IBM Systems and Technology Group, 3039 Cornwallis Road, Research Triangle Park, North Carolina 27709 (kantraj1@us.ibm.com)*. Mr. Kantesaria received a B.S. degree in electrical engineering from the University of Connecticut in 1988 and an M.S. degree in computer science from Rensselaer Polytechnic Institute in 1994. From 1988 to 1997, he was with the Otis Elevator Company, where he worked on new elevator system controller development and test. From 1997 to 1999, he worked at Corning, improving fiber optics manufacturing controls. Since joining IBM in 1999, he has worked on systems management firmware for xSeries systems. Mr. Kantesaria continues to work on service processor firmware for current and next-generation BladeCenter products.

Edward J. Klodnicki *IBM Systems and Technology Group, 3039 Cornwallis Road, Research Triangle Park, North Carolina 27709 (klodnick@us.ibm.com)*. Mr. Klodnicki is a Senior Engineer and has been with IBM for more than 15 years. He received a B.S. degree from the University of Scranton in 1976 and an M.S. degree in electrical engineering from Villanova University in 1984. His previous work at IBM was on the development of the IBM line of automated optical and tape libraries and drives. Mr. Klodnicki is currently developing systems management firmware for service processors and management modules in the xSeries products.

Sumeet Kochar *IBM Systems and Technology Group, 3039 Cornwallis Road, Research Triangle Park, North Carolina 27709 (skochar@us.ibm.com)*. Mr. Kochar is an Advisory Software Engineer and has been with IBM for more than seven years. He received a B.S. degree in physics and an M.S. degree in computer applications from Motilal Nehru National Institute of Technology, Allahabad, India, in 1992 and 1996, respectively. He worked for a year at IBM in India. Most of his career at IBM has been spent in writing firmware for xSeries Intel-based servers. Mr. Kochar developed memory compression diagnostics and is currently the BIOS technical leader for a number of xSeries servers, including processor blades.

Shane M. Lardinois *IBM Systems and Technology Group, 3039 Cornwallis Road, Research Triangle Park, North Carolina 27709 (lardinoi@us.ibm.com)*. Mr. Lardinois received a B.S. degree in computer engineering from the University of Wisconsin in 1995. His IBM experience began in Rochester, Minnesota, working on hardware verification tools for the IBM AS/400. His interests include PowerPC* architecture, video acquisition, and all forms of hardware debug and programming. Mr. Lardinois worked in software until 1997, when he moved to his current position in xSeries systems management firmware.

Carl A. Morrell *IBM Systems and Technology Group, 3039 Cornwallis Road, Research Triangle Park, North Carolina 27709 (cmorrell@us.ibm.com)*. Mr. Morrell is an Advisory Software Engineer and has been with IBM for 23 years. He received

a B.S. degree in computer science from the Rochester Institute of Technology in 1982 and an M.S. degree in computer science from Union College. His prior accomplishments have included the development of a digital signal processing debugger, JPEG image processing routines, and other DSP-related applications. He has worked at designing and implementing embedded applications for the last eight years. Mr. Morrell has several patents pending relating to his current assignment.

Michael S. Rollins *IBM Systems and Technology Group, 3039 Cornwallis Road, Research Triangle Park, North Carolina 27709 (rollinsm@us.ibm.com).* Mr. Rollins received a B.S. degree in electrical engineering from the University of South Alabama in 1996 and an M.S. degree in electrical engineering from the University of Florida in 1999. Before joining IBM, he worked in weapon systems test and research and development as an analyst at Dynetics, Inc. After he joined IBM in early 2000, his work focused on diagnostics for xSeries systems. He is currently a BladeCenter developer. He has several patents either pending or issued. Mr. Rollins is an active Senior Member of the IEEE.

Robert R. Wolford *IBM Systems and Technology Group, 3039 Cornwallis Road, Research Triangle Park, North Carolina 27709 (wolfordr@us.ibm.com).* Mr. Wolford is a Senior Engineer and has been with IBM for 14 years. He holds a B.S. degree in electrical engineering from Pennsylvania State University and has done postgraduate work at Florida Atlantic University. Much of his career has been spent in computer systems development, with an emphasis on video hardware. He is currently working on the high-end 64-bit A Pro IntelliStation* workstation product line. Mr. Wolford holds several patents relating to computer and video architecture and design.

David R. Woodham *IBM Systems and Technology Group, 3039 Cornwallis Road, Research Triangle Park, North Carolina 27709 (dwoodham@us.ibm.com).* Mr. Woodham is an Advisory Software Engineer. He has a B.S. degree in engineering–physics from the University of Oklahoma and a B.S. degree in computer science from the University of Central Oklahoma. He has 17 years of engineering experience in the computer industry, the last five at IBM. His career experience includes developing head–disk interfaces for the disk-drive industry, automated verification of ASIC models for the high-end server industry, and embedded systems management applications for the high-end server industry. Mr. Woodham currently works on the design and development of next-generation BladeCenter products.